

L Number	Hits	Search Text	DB	Time stamp
-	224	du.xa.	USPAT	2004/07/20 13:58
-	9	du.xa. and tap	USPAT	2004/07/14 20:38
-	531	((multiple near3 stage\$1) multistage\$1) with clock	USPAT	2004/07/14 20:59
-	12	(((multiple near3 stage\$1) multistage\$1) with clock) same (delay near5 adjust\$5)	USPAT	2004/07/14 16:11
-	4	multistage adj clock	USPAT	2004/07/14 20:55
-	12	multistage adj clock	USPAT; EPO; JPO; DERWENT; IBM_TDB	2004/07/14 20:55
-	8	(((multiple near3 stage\$1) multistage\$1) with clock) same domino	USPAT	2004/07/14 20:59
-	0	output\$1 with delay\$3 with another	USPAT	2004/07/19 16:11
-	0	delay\$3 with another	USPAT	2004/07/19 16:11
-	0	delay\$3 with "another"	USPAT	2004/07/19 16:10
-	3	another	USPAT	2004/07/19 16:10
-	0	delay\$3 with other	USPAT	2004/07/19 16:11
-	39	output\$1 with delay\$3 with relate	USPAT	2004/07/19 16:34
-	10798	output\$1 with delay\$3 with respect\$7	USPAT	2004/07/19 16:34
-	0	output\$1 with delay\$3 with respect\$7 with each	USPAT	2004/07/19 16:35
-	0	output\$1 with delay\$3 with respect\$7 with other	USPAT	2004/07/19 16:35
-	0	output\$1 with delay\$3 with respect\$7 with another	USPAT	2004/07/19 16:40
-	592	(multiple near3 output\$1) with delay\$3	USPAT	2004/07/19 16:41
-	77	(multiple near3 output\$1) with delay\$3 with (relat\$5 respect\$5)	USPAT	2004/07/19 16:42
-	1	6208907.pn.	USPAT	2004/07/20 13:58

BEST AVAILABLE COPY

## Clock-Delayed Domino for Dynamic Circuit Design

Gin Yee and Carl Sechen

**Abstract**—Clock-delayed (CD) domino is a self-timed dynamic logic family developed to provide single-rail gates with inverting or noninverting outputs. CD domino is a complete logic family and is as easy to design with as static CMOS circuits from a logic design and synthesis perspective. Design tools developed for static CMOS are used as part of a methodology for automating the design of CD domino circuits. The methodology and CD domino's characteristics are demonstrated in the design of a 32-b carry look-ahead adder. The adder was fabricated with MOSIS's 0.8- $\mu\text{m}$  CMOS process with scalable CMOS design rules that allow a 1.0- $\mu\text{m}$  drawn gate length. Measurements of the adder show a worst case addition of 2.1 ns. The CD domino adder is 1.6× faster than a dual-rail domino adder designed with the same cell library and technology.

**Index Terms**—Adder, dynamic logic circuit, dynamic logic clocking, inverting single-rail dynamic gates, self-timed circuits.

### I. INTRODUCTION

Dynamic circuits have become necessary for designing high-speed and compact circuits, as seen by their use in microprocessors [1], [2]. The reduced input capacitance and use of only nMOS logic transistors make domino circuits faster and smaller than their static counterparts [3]. One of dynamic logic's major shortcomings is its monotonic nature. This restriction causes domino logic, a widely used dynamic logic family, to allow only noninverting functions. However, current logic synthesis tools and most practical circuit designs require the flexibility to use inverting and noninverting functions in any combination. While dynamic logic with inverting outputs are known, they generate both polarities of the output in differential or dual-rail fashion or use latches [4]–[6].

Clock-delayed domino provides any logic function through the use of a self-timed delay-matched clock tree for the precharge and evaluation clock used in dynamic circuits. The delayed clocks are set to always arrive after the data inputs to a dynamic gate have settled. A delayed clock is used in wave-domino logic for wave-pipelining, but was not considered for providing single-rail inverting or noninverting dynamic logic gates [7]. Footless domino uses delayed clocks to reduce the through-current between power and ground [1]. Delayed evaluation was used for a multiplier design, and delayed precharge was used for reducing power consumption and noise during the precharge phase of domino circuits [8], [9]. Delay matching for self-timed circuits was used in an SRAM design [10].

A 32-b carry look-ahead adder (CLA) was designed using CD domino to demonstrate its design flexibility and speed advantages. The CLA logic equations were rewritten to take advantage of high fanin dynamic NOR and OR gates, and a dynamic XOR gate.

The following section describes the key characteristics of CD domino. Section III describes a design methodology and tools for

Manuscript received April 24, 1998; revised April 25, 1999. This work was supported by the Center for the Design of Analog/Digital IC's (CDADIC), Compaq Computer Corporation, the National Science Foundation, the Semiconductor Research Corporation, Sun Microsystems, Inc., and the Tau Beta Pi Association.

G. Yee is with Sun Microsystems, Palo Alto, CA 94303 USA, and also with the Department of Electrical Engineering, University of Washington, Seattle, WA 98195-2500 USA (e-mail: gsyee@twolf14.ee.washington.edu).

C. Sechen is with the Department of Electrical Engineering, University of Washington, Seattle, WA 98195-2500 USA (e-mail: sechen@twolf14.ee.washington.edu).

Publisher Item Identifier S 1063-8210(00)01024-6.

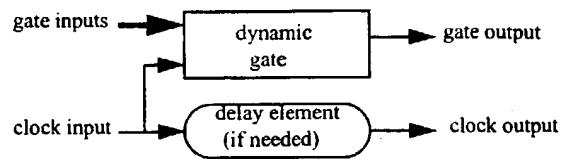


Fig. 1. CD domino gate with a dynamic gate and delay element.

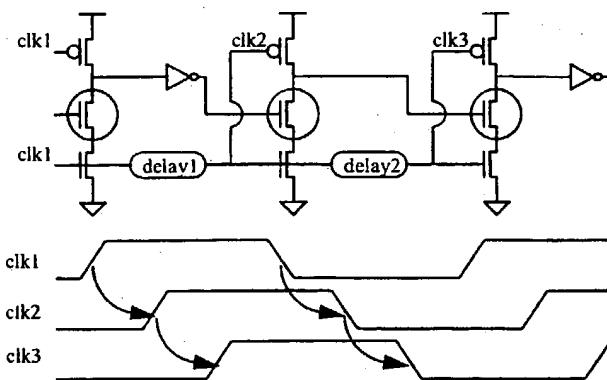


Fig. 2. CD domino path.

CD domino. Next, the design flexibility and speed advantages of CD domino are demonstrated in the design of a 32-b CLA adder in Section IV. This is followed by the adder's measurement results in Section V and the conclusion.

### II. CLOCK-DELAYED DOMINO LOGIC

#### A. CD Domino Operation

CD domino addresses the logic and circuit design difficulties of dynamic circuits by providing any logic function without restrictions on how the gates are used together. This is accomplished through the use of self-timed delays for the precharge and evaluation clocks. Each CD domino gate consists of a dynamic gate and, if necessary, a delay element, as shown in Fig. 1. It is similar to the self-timed scheme using delay matching as defined in [11]. The self-timed clock output of the delay element tells the next gate when the data output is ready, as shown in Fig. 2. Thus, the delay set by the delay element is always greater than the worst case delay of the dynamic gate, plus a margin.

It is critical that the clock output rising edge occurs after the gate output has switched and never before. The delay elements will always be the critical path, and the data hazard in dynamic logic caused by non-monotonic input transitions during the evaluation phase is prevented. This self-timed scheme for CD domino uses single-rail dynamic gates, which is in contrast to the well-known dual-rail self-timed scheme shown in Fig. 3 [12].

#### B. CD Domino Gates

Noninverting CD domino gates are simply domino gates as described in [3]. Inverting CD domino gates can be designed by removing or adding an inverter to a domino gate, as shown by the second gate in Fig. 2. An inverting CD domino gate with its output precharged low is shown in Fig. 4. While not necessary, having the output precharged low for all dynamic gates maintains the monotonic nature of dynamic gates and prevents a precharge glitch possible with precharge high dynamic

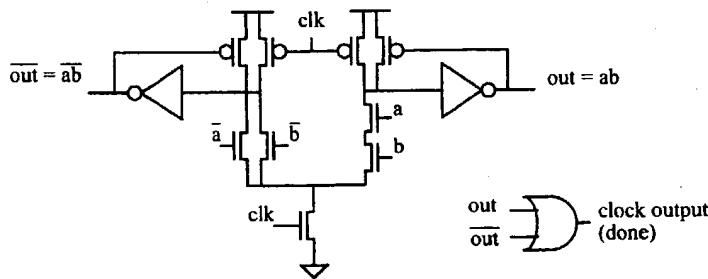


Fig. 3. Dual-rail self-timed circuit.

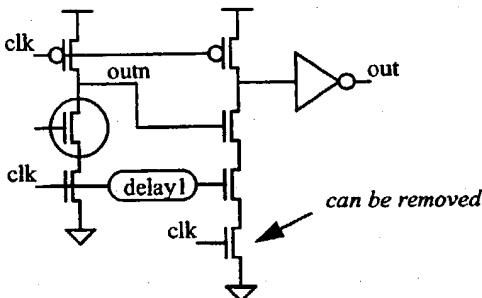


Fig. 4. An inverting dynamic gate that precharges low.

gates. The delay element used in the inverting CD domino gate shown in Fig. 4 is set to the delay of the nMOS pull-down network plus a margin to provide correct operation.

#### C. Delay Element for Delay Matching

A delay element, shown in Fig. 5, is used to create a delay chain for the delayed clocks used to precharge and evaluate the dynamic gates. The delay of the delay element and the rest of the delay circuit consists of four components matched to its corresponding dynamic gate: the intrinsic gate delay, the output net wiring delay, the fanout gate load delay, and a margin. The gate delay is set equal to the worst case pull-down delay of the corresponding dynamic gate during the evaluation phase. Matching the gate delay can be done using a dummy domino gate. A margin is added to account for setup times to the next gate, variations in fabrication processes, voltage, and temperature variations between the delay element and its gate, and differences in the signal delay due to output wiring, fanout load, and coupling parasitics. A margin of at least 20% of the gate delay was added to the delay elements for the adder circuit to insure proper precharge and evaluation of the CD domino gates.

#### D. Simplest CD Domino Clocking Scheme

For CD domino circuits, two clocking schemes are possible. In the most basic scheme, only the slowest gate at each gate level needs to have a delay element. The clock output from the previous gate level would be used by the gates at the next level. This scheme, shown in Fig. 6, is similar to the wave-pipelining method used by Lien and Burleson for wave-domino [7].

The primary outputs provided by this clocking scheme must wait for the slowest gate at each gate level. Thus, the performance of the simplest CD domino clocking scheme is based on the slowest gate at each gate level and not necessarily the critical path.

#### E. General CD Domino Clocking Scheme

The second clocking scheme is based on a more general self-timed delay tree obtained by having each gate use the clock output from

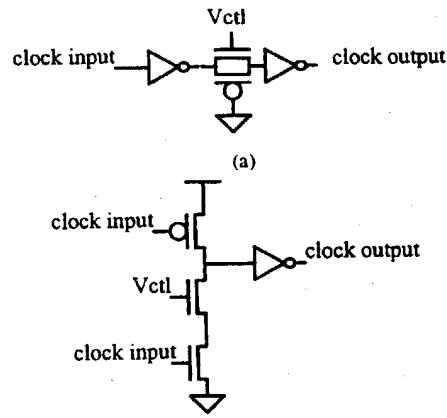


Fig. 5. Example adjustable delay element.

the gate of its slowest input, rather than the same clock for the entire gate level. As Fig. 7 shows, this general approach shifts away from the timing constraints in pipelining. At the cost of a few extra delay elements, the outputs of the circuit evaluate faster because each gate does not have to wait for the slowest gate's clock output from the previous gate level, just the clock output from its slowest input. If a gate's output is not the slowest input to any gate, then it does not need a delay element.

### III. METHODOLOGY AND TOOLS FOR CD DOMINO CIRCUITS

While synthesis and design tools have been developed for standard domino, they operate with the constraint of using dual-rail or differential domino logic [13]–[15]. By providing any logic function, CD domino gates can be used by design tools developed for static CMOS gates by simply replacing the static CMOS library with a CD domino library. After logic synthesis or handcrafted design with a tool designed for static circuits, delay elements are inserted into the circuit netlist to provide the correct self-timed delays, and a proper precharge-evaluation clock. Thus, combinational logic blocks can be designed having the speed of dynamic logic using CD domino, but without the functional limitations or higher area and power costs found in other dynamic logic families.

A design methodology developed for CD domino circuits is shown in Fig. 8. Cell and delay element library design has been described in earlier sections. Library characterization is self explanatory. The other steps are described as follows.

#### A. Gate Netlist

The methodology allows the gate netlist to be handcrafted for custom designs, or generated by a synthesis tool. Any synthesis tool developed

BEST AVAILABLE COPY

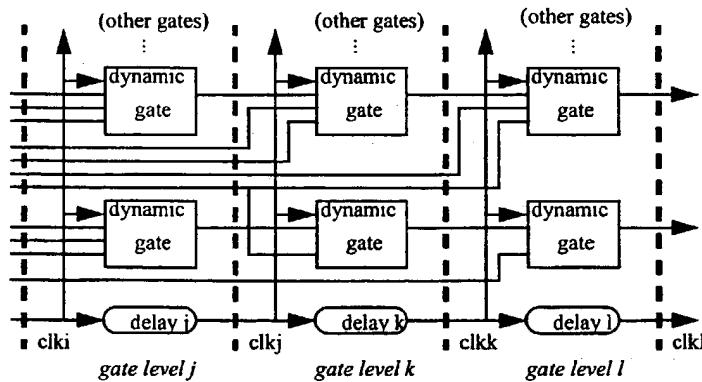


Fig. 6. Simplest clocking scheme for CD domino.

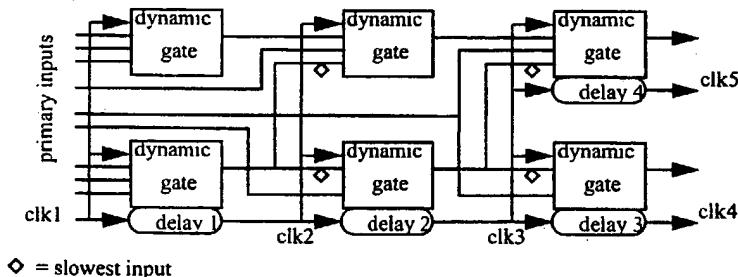


Fig. 7. General clocking scheme for CD domino.

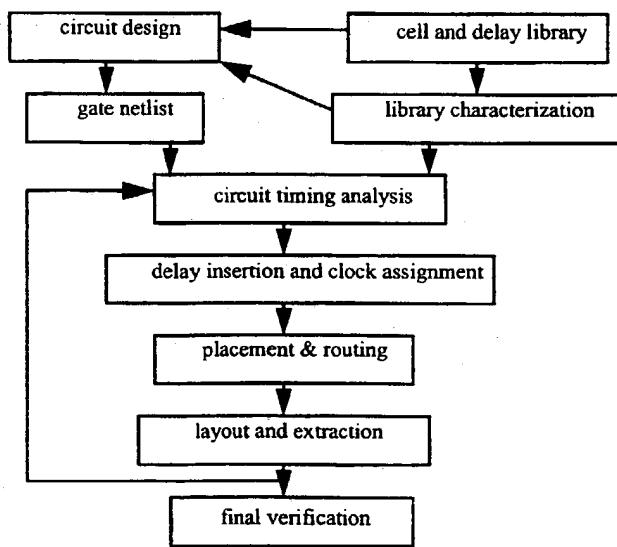


Fig. 8. Design methodology for CD domino.

for static CMOS circuits can be used for this step. The library given to the synthesis tool is still "static" as far as the tool is concerned. The key difference is that each gate in the "static" library corresponds directly to a gate in the dynamic library and has the same functionality. Thus, at this point, the netlist is still a static CMOS netlist. Care must be taken to allow the synthesis tool to fully take advantage of the fast high fanin dynamic gates available in the library.

```

levelize_netlist()
  set flops to gate_level = 0
  for each gate with primary inputs or flop inputs
    gate_level = 1
  for i=1 to maximum_gate_level
    for each gate
      if all inputs are from <= level i
        gate_level = i + 1

timing_analysis()
  levelize_netlist()
  for gates at level i=1 to max_gate_level
    if i=1
      gate_path_delay = gate_delay
    else
      find_slowest_input()
      gate_path_delay = slowest_input->gate_path_delay + gate_delay

```

Fig. 9. Algorithm for gate levelization and timing analysis.

### B. Circuit Timing Analysis

A circuit timing analysis tool was developed for the CD domino methodology. Using the characterized gate and delay element delay information in a lookup table, a static timing analysis is done on the netlist by leveling the netlist (assigning a gate level to each gate), and adding up the path delays to each gate. The netlist levelizing and timing analysis algorithms are given in Fig. 9. The timing analysis tool simply adds up the delays of the gates along all paths starting with gates at the first gate level. More accurate delay estimation is made using the extracted layout information fed back to the timing analysis tool. As a final check, the tool also writes out a critical path SPICE simulation with the extracted parasitics, delay elements, and gates.

**BEST AVAILABLE COPY**

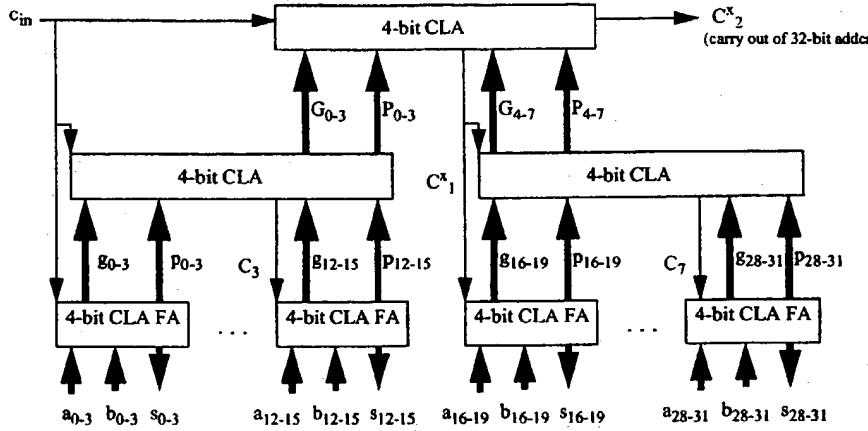


Fig. 10. Block diagram of static and domino 32-b CLA adder.

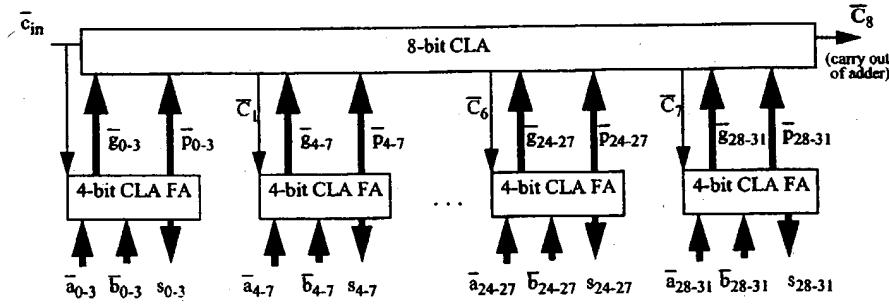


Fig. 11. Block diagram of CD domino adder.

### C. Delay Element Insertion and Clock Assignment

The timing analysis data is provided to the delay element insertion tool to generate the clocking network depending on the user option for the simplest clocking scheme or the general scheme described in Section II. The tool also pads out short paths with dynamic buffers (1-input OR) to ensure enough evaluation time overlap between delayed clocks. For example, a problem can arise if a gate with  $\text{clk}_1$  drives a gate with  $\text{clk}_9$  and they do not have any overlap in their clocks' evaluation phase. The number of delayed clock phases that a signal can skip is a user input.

Once the delay elements have been chosen according to the clocking scheme, the delayed clock phases are assigned to each gate. For the simplest clocking scheme, clock assignment is determined by each gate's gate level, as determined by `levelize_netlist()`. For the general clocking scheme, each gate gets its clock from its slowest input, which will have a delay element associated with it. The static netlist is now converted to a dynamic gate netlist by directly mapping the gates to a dynamic library, and assigning the clocks according to the user defined clocking scheme.

### D. Placement and Routing

Placement and routing of the design is done using the standard cell library and dynamic gate netlist. The skew between each clock phase is reduced by restricting the placement, and is done using industry placement tools. For example, with the simplest clocking scheme, gates at the same level are placed in the same rows to reduce the wire length of the clock net. After extraction, the parasitic data is fed back to the timing analysis tool for more accurate delay calculation.

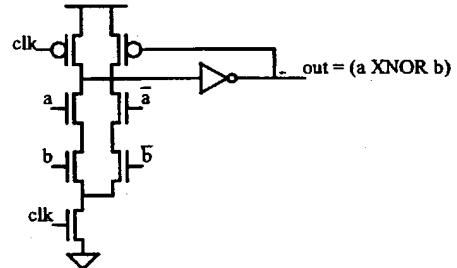


Fig. 12. Schematic of dynamic 2-input XNOR.

## IV. ADDER DESIGN

The carry lookahead (CLA) adder design was chosen to demonstrate the use of CD domino gates and the design methodology described in Section III.

### A. Static and Dual-Rail Adder Design

When using static CMOS or standard domino logic, a common 32-b CLA adder design uses eight 4-b full adder (FA) blocks with two CLA logic levels, as shown in Fig. 10. The CLA FA logic equation for the first level is given by (1), while (2) and (3) provide the CLA logic equations for the second and third levels [16]

$$\begin{aligned}
 g_i &= a_i b_i, \quad p_i = a_i \oplus b_i, \quad s_i = p_i \oplus c_i \\
 c_0 &= c_{\text{in}}, \quad c_1 = g_0 + p_0 c_{\text{in}}, \\
 c_2 &= g_1 + p_1 g_0 + p_1 p_0 c_{\text{in}}, \\
 c_3 &= g_2 + p_2 g_1 + p_2 p_1 g_0 + p_2 p_1 p_0 c_{\text{in}}
 \end{aligned} \tag{1}$$

BEST AVAILABLE COPY

TABLE I  
SIMULATED ADDER COMPARISONS IN MOSIS' 0.8- $\mu\text{m}$  TECHNOLOGY

32-bit adder type	Simulated worst case delay [ns]	Adder core area [ $\text{mm}^2$ ]	Avg. power [W]	Power-delay product [W-ns]
Static CMOS	6.97	0.565	0.0756	0.527
Dual-Rail	4.23	0.923	0.1994	0.845
CD Domino	2.73	0.858	0.1733	0.473

$$\begin{aligned}
 P_0 &= p_3 p_2 p_1 p_0, \dots, P_3 = p_{15} p_{14} p_{13} p_{12} \\
 G_0 &= g_3 + p_3 g_2 + p_3 p_2 g_1 + p_3 p_2 p_1 g_0, \dots \\
 G_3 &= g_{15} + p_{15} g_{14} + p_{15} p_{14} g_{13} + p_{15} p_{14} p_{13} g_{12} \\
 C_1 &= G_0 + P_0 c_{\text{in}}, \quad C_2 = G_1 + P_1 G_0 + P_1 P_0 c_{\text{in}} \\
 C_3 &= G_2 + P_2 G_1 + P_2 P_1 G_0 + P_2 P_1 P_0 c_{\text{in}} \quad (2) \\
 P_0^x &= P_3 P_2 P_1 P_0, \quad P_1^x = P_7 P_6 P_5 P_4 \\
 G_0^x &= G_3 + P_3 G_2 + P_3 P_2 G_1 + P_3 P_2 P_1 G_0 \\
 G_1^x &= G_7 + P_7 G_6 + P_7 P_6 G_5 + P_7 P_6 P_5 G_4 \\
 C_1^x &= G_0^x + P_0^x c_{\text{in}}, \quad C_2^x = G_1^x + P_1^x G_0^x + P_1^x P_0^x c_{\text{in}}. \quad (3)
 \end{aligned}$$

The design in Fig. 10 uses 4-b FA blocks because of the speed limitations imposed by the number of series transistors required for higher carry bit logic in standard domino and static CMOS logic. In a purely standard domino CLA design, the dual-rail approach must be used to provide both polarities of the inputs needed by the XOR gates ( $s_i$  and  $p_i$  functions). The dual of (1)–(3) are not shown since they can be easily derived by inverting all inputs, swapping AND gates for OR gates, XNOR gates for XOR gates, and vice versa.

### B. CD Domino Adder Design

Using CD domino gates, a CLA adder was designed to demonstrate the use of inverting and fast high fanin dynamic gates. The adder uses eight 4-b FA blocks with a single level of CLA logic, as shown in Fig. 11. The 8-b CLA logic block provides the lookahead carry values ( $\bar{C}_i$ ) for the eight FA blocks. The CLA FA logic equations are given by (4). They are the same as in (1) except inverted to take advantage of wide NOR gates instead of AND gates using DeMorgan's Law. Likewise, the 8-b CLA logic block equations are given by (5) and also inverted to use wide NOR gates. Although inputs and intermediated nodes are all inverted, the final sum outputs are not inverted, and both polarities of  $g_i$  and  $G_i$  are provided with static inverters. An 8-b CLA logic block is practical because of the fast high fanin OR and NOR gates possible with dynamic logic. This feature of CD domino improves the performance of the adder design, but would not be practical with static CMOS or dual-rail domino

$$\begin{aligned}
 \bar{g}_i &= \bar{a}_i + \bar{b}_i, \quad \bar{p}_i = \bar{a}_i \odot \bar{b}_i, \quad s_i = \bar{p}_i \oplus \bar{c}_i \\
 \bar{c}_0 &= \bar{c}_{\text{in}}, \quad \bar{c}_1 = g_0 + (\bar{p}_0 + \bar{c}_{\text{in}}), \dots \\
 \bar{c}_3 &= g_2 + (\bar{p}_2 + \bar{g}_1) + \dots + (\bar{p}_2 + \bar{p}_1 + \bar{p}_0 + \bar{c}_{\text{in}}) \quad (4) \\
 \bar{P}_0 &= \bar{p}_3 + \bar{p}_2 + \bar{p}_1 + \bar{p}_0, \dots \\
 \bar{P}_7 &= \bar{p}_{31} + \bar{p}_{30} + \bar{p}_{29} + \bar{p}_{28} \\
 \bar{G}_0 &= g_3 + (\bar{p}_3 + \bar{g}_2) + \dots + (\bar{p}_3 + \bar{p}_2 + \bar{p}_1 + \bar{g}_0), \dots \\
 \bar{G}_7 &= g_{31} + (\bar{p}_{31} + \bar{g}_{30}) + \dots + (\bar{p}_{31} + \bar{p}_{30} + \bar{p}_{29} + \bar{g}_{28}) \\
 \bar{C}_1 &= G_0 + (\bar{P}_0 + \bar{c}_{\text{in}}), \dots \\
 \bar{C}_8 &= G_7 + (\bar{P}_7 + \bar{G}_6) + \dots + (\bar{P}_7 + \bar{P}_6 + \dots + \bar{P}_0 + \bar{c}_{\text{in}}). \quad (5)
 \end{aligned}$$

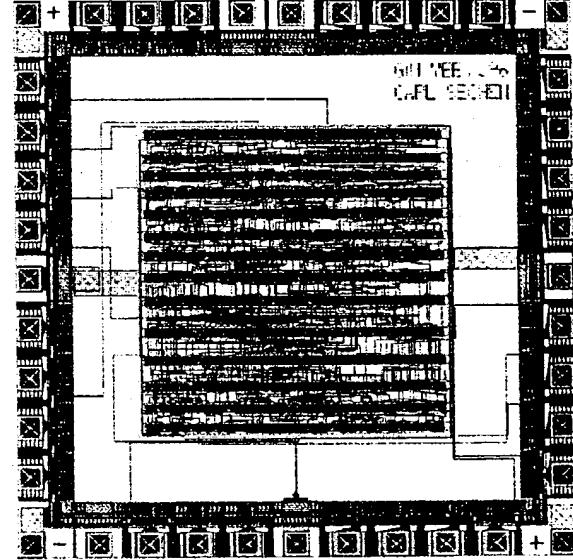


Fig. 13. CD domino adder chip layout.

Because inverting and noninverting gates can be used together, compact dynamic XOR and XNOR gates are used in the design without requiring a dual-rail approach for CD domino. Fig. 12 shows the schematic for the dynamic XNOR gate used in the dynamic adder designs.

The performance improvements obtained using CD domino are primarily due to the reduction in the number of gate levels. Using high fanin NOR and OR gates allows the collapsing of logic cones into fewer levels. In more conventional logic families such as static CMOS or dual-rail domino, it is not possible to use very wide gates (such as eight input NOR's and OR's) due to the nominal limit on the number of series transistors (typically three or four, and even at four the edge rates are degraded). Significant reductions in the number of gate levels translates to large performance improvements even though a 20% margin is designed into the delay elements. For the CD domino adder chip, programmable delay elements were used to allow adjustable delays after fabrication, as shown in Fig. 5. A global external input voltage,  $V_{ctl}$ , is routed to all delay elements to control the delay through the delay elements and increase or decrease the margin. For all of the work reported here,  $V_{ctl}$  was set at  $VDD$  to give a nominal margin of at least 20%. Reducing  $V_{ctl}$  below  $VDD$  will only slow down the delay element and increase the margin, if needed.

### V. ADDER RESULTS

Static CMOS and standard domino versions of the 32-b CLA adders, described in the previous section, were designed and compared to the CD domino adder. The dual-rail domino and CD domino adders were

BEST AVAILABLE COPY

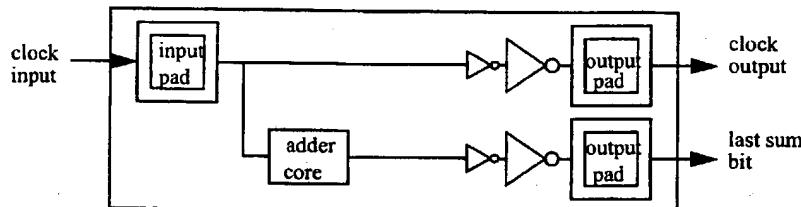


Fig. 14. Circuit used to measure adder delay within chip core.

TABLE II  
MEASURED ADDER RESULTS FROM MOSIS 0.8- $\mu$ m ADDER CHIPS

32-bit adder type	Measured worst case delay [ns]	Adder core area [ $\text{mm}^2$ ]	Delay-area [ns-mm $^2$ ]
Dual-rail domino	3.4	0.923	3.14
CD domino	2.1	0.858	1.80

designed using the same standard cell library of dynamic gates. A separate library was developed for the static adder. The designs were routed with three metal layers using MOSIS' scalable CMOS 0.8- $\mu$ m process, which allows 1.0- $\mu$ m drawn gate lengths. Both libraries were developed with speed and area considerations. Table I compares the simulated results from extracted netlists for the worst case 32-b addition, the total area of the adder layout cores, and the power-delay product. The CD domino adder is 1.6× faster than the dual-rail domino design, while requiring 8% less area. The simulation and measured delay results reported for CD domino have the margin of at least 20% designed into the delay elements. The CD domino adder had a power-delay product better than the static adder by over 10%. The CD domino adder chip is shown in Fig. 13. The circuit diagram shown in Fig. 14 was used to obtain the worst case adder delays within the chip core. Table II provides the measured results for the dual-rail and CD domino adders.

## VI. CONCLUSION

A self-timed dynamic logic family, an automated methodology, and tools for designing dynamic circuits with it were presented. CD domino provides any logic function allowed by static CMOS, as well as fast high fanin gates that are not possible with static circuits or standard domino logic. As the adder comparisons in the previous section show, the CD domino implementation of the 32-b CLA adder has significant speed improvements over the static CMOS and dual-rail domino counterparts. Measurements from fabricated chips show the CD domino adder to be more than 1.6× faster than the dual-rail domino adder using the same gate library, and requiring 8% less area.

The performance and functionality advantages are possible because CD domino provides fast high fanin dynamic gates while maintaining the flexibility in design of static CMOS. The use of higher fanin gates does not hurt the speed of each gate and improves overall performance by reducing the number of logic levels in the critical path. Just as important is the significant reduction in the design time of the self-timed dynamic circuits made possible by the methodology and design automation tools developed for CD domino. Fast and robust CD domino circuits can be designed with practical delay element margins.

## REFERENCES

- [1] R. Colwell and R. Steck, "A 0.6 mm BiCMOS processor with dynamic execution," *ISSCC Tech. Dig.*, pp. 176–177, Feb. 1995.
- [2] A. Charnas *et al.*, "A 64 b microprocessor with multimedia support," *ISSCC Tech. Dig.*, pp. 177–179, Feb. 1995.

- [3] R. H. Krambeck, C. M. Lee, and H. S. Law, "High-speed compact circuits with CMOS," *IEEE J. Solid-State Circuits*, vol. SC-17, pp. 614–619, June 1982.
- [4] L. Heller and W. Griffin, "Cascade voltage switch logic: A differential CMOS logic family," *ISSCC Tech. Dig.*, pp. 16–17, 1984.
- [5] N. Goncalves and H. De Man, "NORA: A racefree dynamic CMOS technique for pipelined logic structures," *IEEE J. Solid-State Circuits*, vol. SC-18, pp. 614–619, June 1983.
- [6] C. Wu and K. Cheng, "Latched CMOS differential logic (LCDL) for complex high-speed VLSI," *IEEE J. Solid-State Circuits*, vol. 26, pp. 1324–1328, Sept. 1991.
- [7] W. Lien and W. P. Burleson, "Wave-domino logic: Theory and applications," *IEEE Trans. Circuits Syst.*, vol. 42, pp. 78–91, Feb. 1995.
- [8] G. Sobelman and D. Raatz, "Low-power multiplier design using delayed evaluation," in *Proc. IEEE Int. Symp. Circuits Syst.*, May 1995, pp. 1564–1567.
- [9] G. Boudou, "Staggering of the precharge clocks in domino circuits," *IBM Tech. Discov. Bull.*, vol. 27, pp. 4340–4341, Dec. 1984.
- [10] T. Chappell, B. Chappell, S. Schuster, J. Allan, S. Klepner, R. Joshi, and R. Franch, "A 2 ns cycle, 3.8 ns access 512-kb CMOS ECL SRAM with a fully pipelined architecture," *IEEE J. Solid-State Circuits*, vol. 26, pp. 1577–1585, Nov. 1991.
- [11] C. Mead and L. Conway, *Introduction to VLSI Systems*. Reading, MA: Addison-Wesley, 1980.
- [12] T. Williams and M. Horowitz, "A zero-overhead self-timed 160-ns 54-b CMOS divider," *IEEE J. Solid-State Circuits*, vol. 26, pp. 1651–1661, Nov. 1991.
- [13] G. De Micheli, "Performance-oriented synthesis of large-scale domino CMOS circuits," *IEEE Trans. Computer-Aided Design*, vol. CAD-6, pp. 751–765, Sept. 1987.
- [14] M. Hoffmann and A. Newton, "A domino CMOS logic synthesis system," in *Proc. IEEE Int. Symp. Circuits Syst.*, 1985, pp. 411–414.
- [15] R. Puri, A. Bjorksten, and T. Rosser, "Logic optimization by output phase assignment in dynamic logic synthesis," in *Proc. IEEE Int. Conf. Computer-Aided Design*, 1996, pp. 2–8.
- [16] N. Weste and K. Eshraghian, *Principles of CMOS VLSI Design*. Reading, MA: Addison-Wesley, 1993.

BEST AVAILABLE COPY